Chapter 6 INTEGRATED FRAMEWORK FOR TRAFFIC ENGINEERING AND FAILURE RESTORATION

Unexpected equipment failures can cause stable and secure routes to become unstable which can lead to performance degradation and unstable network. To handle traffic engineering in presence of failures, a framework is proposed in this chapter that optimizes load balancing for wide range of failure layouts.

6.1 Elementary Network Framework

The framework proposed here uses elementary, standard routers to balance load before, during and after failures. These routers reserves most of the functionality in mainframe organization which implements offline optimization. The network mainframe organization determines discrete routes between each pair of edge routers and informs each entering router how to divide traffic over these routes under a range of failure layouts. Each edge router tracks route-level failure and uses this knowledge to regulate the dissolution of traffic over the left-over routes.

6.2 Characteristics of Framework

The new framework being proposed here has the following characteristics:

(i) Pre-estimated multipath routing: In this routers do not need to evaluate routes again & again because multiple routes are constructed in advance between each pair of edge routers. This reduces load on router and enhances route stability.

(ii) Failure Tracking: The routers located at the entrance of routes perform failure restoration based only on which routes have failed. A smallest control plane performs failure tracking across failed links. Due to this simple and cheap routes can be selected.

(iii) Bounded Transformation to route failures: Once the failed links/routes are tracked, the entering router counterbalances the traffic on the remaining routes, based on which routes failed. This refrain the routers from circulating real-time updates about link load. By doing this inconsistency can be avoided.
6.2.1 Pre-estimated Multipath Routing

Many existing routing protocols calculate a single path between each router pair and alter that path in return to alteration in topology. But dynamic routing has many drawbacks which includes overhead on routers (to distribute topology information and calculate paths) and the temporary interruption during convergence. When convergence is made faster, it will lead to increased complexity of routing software and more overhead on routers, because it will distribute more information or update it faster. So instead of reducing convergence time or utilize other mechanisms to trace temporary loops, it is better not use dynamic routing. So framework proposed here does not use dynamic routing. Instead it uses numerous pre-estimated routes between each pair of edge routers. This grants the entering routers to reconcile to failures by transferring traffic away from failed links. Routers do not use failed links till they start working again. This lessens router software complexity and overhead on protocols.

The mainframe organization determines the routes which are based on both traffic engineering and failure restoration and place these routes in the elementary routers. The mainframe organization can pick miscellaneous routes that certify connectivity in case of failures in addition to several mutual failures. Use of multiple routes leads to improved load balancing [5, 47] whether or not failures have occurred. The framework proposed here concedes supple bifurcation of traffic over multiple routes. Anyhow it does not depend upon dynamic transformation of the traffic bifurcation. Rather the entering router has an intelligible static composition that regulates the bifurcation of traffic over the accessible routes, at the same time mid-way routers solely move packets over pre-estimated routes. The mainframe organization optimizes this arrangement immediately based on predicted traffic & failures. This all reduces protocol overhead and increases steadiness in load-sensitive routing protocols. Also mainframe organization can utilize information about cluster risks and expected traffic requests- information which routers do not have.

6.2.2 Failure Tracking

Maximum numbers of routing protocols track down failures by switching Hello messages between neighbouring routers and broadcast the modified topologies in the network. Such kind of failure tracking measure depends upon small timers which puts extra pressure on the routers. Further, many failures are provoked by proposed maintenance [6] which can lead to confluence situations; one of which is for the link failure and another for the restoration that
both cause temporary interruptions. Moreover, the Hello messages do not track all kinds of failures such as misconfigurations and malicious attacks.

The framework proposed here depends upon route level failure tracking:

- In this kind of tracking, each entering-exiting router pair has a dialogue to supervise each of its routes (e.g.: as in BFD [19]).
- The explorer packet can be piggybacked on current data traffic which restrains the need of “Hello” messages when route is carrying regular data traffic.
- The above steps help in fast failure tracking without the extra load of explorer traffic. The above mentioned method presents a more practical view of the accuracy of a route, since the packets vary in size, addresses and so on.
- Moreover the packets are managed by the hardware interfaces, so they do not consume processing resources at mid-way routers.

Even though entering router does not know which linked failed, it will have information about route failures which is competent to obviate the failed route. Actually, since the routers do not need to be familiar of topology so no control protocol is needed to interchange topology information.

Moreover only those entering routers need information about route failures that have routes which pass through failed edge, other routers do not need such information. But mainframe organization should be aware of modified topology, so that failed route can be repaired. But this tracking problem can be dealt on much longer timescale since it does not affect failure-restoration time for data traffic.

### 6.2.3 Bounded Transformation to Route failures

Although router is a simple device that does not take part in a routing protocol, assemble congestion observation or resolves any complicated problems but it plays very significant role in circulation of traffic in case of link failures or restoration. There are two different ways in which router can bifurcates traffic on working links:

i) Case-Debased bifurcation

ii) Case-Autonomous bifurcation
6.2.3.1 Ideal Load Balancing

In this ideal routes and bifurcation rates are determined in dependently for each possible failure state. This technique results in best feasible load balancing [35]. This technique cannot be implemented in real world scenario because routers must have knowledge about state of links and attain information about all link failures. Even on links the router’s paths do not traverse. Hence this solution defies the proposed framework, but still this solution is appealing because it supports a lower bound on the magnitude of congestion attainable by other two solutions.

6.2.3.2 Case-Debased Bifurcation

In this solution every entering router has an independent anatomy entry with route-bifurcation weights for each consolidated route failures to a distinct exiting router. For e.g.:

- Assume a router has 4 routes to an exit router.
- Router anatomy encompasses 15 entries i.e. 1 entry for each of \((2)^{4-1}\) aggregate of route failures.
- For each anatomy entry, mainframe organization has determined 3 weights in advance. These weights are one per route, with 0 for any failed routes.
- After tracking route failure, the entering router scrutinizes a pre-estimated table to elect the relevant weights for bifurcation of traffic directed to the exiting router.

6.2.3.3 Case-Autonomous Bifurcation

This solution reduces complexity in the router anatomy. It provides single set of weights over all failure layouts. This will result in simple weight table i.e. if an entering router has 3 routes to exiting router then entering router will have only 3 weights, one for each route. If any route fails, then entering router simply regulates the traffic on remaining routes. So this will result in minimum robust optimization of restricted anatomy parameters by mainframe organization to get good load balancing performance over a range of failure layouts.

6.3 Network Inclusive Decisions

In proposed framework, network mainframe organization performs network inclusive decisions to estimate routes and traffic bifurcation range that counterbalance load adequately over a range of failure layouts. Sections 6.3.1, 6.3.2, 6.3.3 and 6.3.4 discusses about the
information that mainframe organization must have about network topology, traffic requests and common risks. Sections 6.3.5 and 6.4 discuss the computation of various miscellaneous routes and traffic bifurcation range for both Case-Debased and Case-Autonomous bifurcation by the mainframe organization.

6.3.1 Network Management Metrics

The mainframe organization uses following parameters to estimate routes and bifurcation ratio:

\( G (V, E) \) - Network with vertices V and directed edges E

\( M^a \) - Magnitude of edge ‘a’ where \( a \in E \)

\( PL^a \) - Procreation lag on edge ‘a’ where \( a \in E \)

\( GS \) - Group of network failure states

\( gs \) - Group of failed links

\( W^{gs} \) - Weight of failed links \( gs \) where \( gs \in GS \)

\( RQ \) - set of requests

\( x^q \) - Source of request where \( q \in RQ \)

\( y^q \) - Destination of request where \( q \in RQ \)

\( b^q \) - Flow requirement of request where \( q \in RQ \)

\( R_q \) - Route available to request where \( q \in RQ \)

\( \beta_r \) - Fraction of request assigned to route ‘r’

\( TG_q \) - Group of tangible failure states for node \( x^q \)

\( tg (gs) \) - State tangible by \( x^q \) in failure state \( gs \in GS \)

\( R^{tg}_q \) - Routes available to \( x^q \) in failure state

\( T_{rg}^{gs} \) - Traffic on route ‘r’ in failure state \( gs \in GS \)

\( T_{rg}^{tg} \) - Traffic on route ‘r’ in failure state \( tg \in TG \)
$TF^{gs}_a$ - Total flow on edge ‘a’ in failure state ‘gs’

$TF^{gs}_{a,q}$ - Flow of request ‘q’ on edge ‘a’ in failure state ‘gs’

6.3.2 Static Topology

The mainframe organization makes decisions which depend on the topology of network—the routers and the links that have been implemented. The topology can be represented by graph $G (V, E)$ where ‘$V$’ is set of vertices and ‘$E$’ is set of directed edges. Each edge can have magnitude represented as $M^a$ where $a \in E$ and edge can also have procreation lag as $PL^a$ where $a \in E$.

6.3.3 Aggregation of Common Link Risks

The mainframe organization should have knowledge about common link failure risks such as connecting to some line card or router or crossing over the same optical fiber or amplifier [37]. These group common risks are denoted by $GS$. $gs \in GS$ is a set of edges that may fail together. For example: a router failure is depicted by set of its corresponding links, a fiber cut is depicted by all links in the damaged fiber bundle and failure-free case is depicted by empty set $\phi$. The mainframe organization must also have estimated results from previous failures. The weight represents $W^{gs}$ the probability or signification of failure state ‘GS’.

6.3.4 Conventional Traffic Requests

The mainframe system must have knowledge about the predictable traffic requests which should be based on the previous estimated and forecasted traffic transformation. Each traffic request $q \in RQ$ is represented by $(x^q, y^q, b^q )$ where $x^q \in V$ is traffic request source (entering router), $y^q \in V$ is traffic request destination (exit router) and $b^q$ is flow requirement of request (measured traffic). For clarity, it is supposed that all requests remain attached for each failure layout rather a request can be excluded for each failure case that detaches it. In usual procedure, the mainframe organization may have a time series of traffic requests (e.g.: for different hours in day) and optimize the network anatomy across all these requests.

The output of mainframe organization is set of routes $R_q$ for each request $q$ and the bifurcation range for each route. In each group of failed links ‘$gs$’, the traffic bifurcation by entering router $x^g$ rely upon only which routes have failed and not on which failure layout ‘$gs$’ has occurred. In fact various failure layouts may affect the same subset of routes in $R_q$. 
To manipulate distinct request $q$, consider a set of $TG_q$ which represents set of tangible failure states where each tangible state $tg \in TG_q$ coincide to an appropriate $R_q^{tg} \subset R_q$ depicting the accessible routes.

The amount of flow assigned to route $r$ in tangible failure state $tg \in TG_q$ is $T_r^{tg}$. The total flow on edge $a$ in failure state $gs$ is $TF_a^{gs}$ and flow on edge $a$ corresponding to request $q$ is $TF_a^{gs,q}$.

The task of mainframe organization is to estimate routes and bifurcation range which reduces congestion over the range of available failure states. The characteristics traffic engineering objective is to minimize

$$\sum_{a \in E} \phi \left( \frac{TF_a^{gs}}{M^a} \right) \tag{1}$$

Where $\phi$ is a convex function of link load which penalizes the most congested links while still accounting for load on the remaining links.

The final objective function to minimize congestion across failure layouts is

$$\text{Obj} \left( \frac{TF_a^{gs1}}{M^{a1}} \right) = \sum_{gs \in GS} W^{gs} \sum_{a \in E} \phi$$

Minimizing the objective function is goal of all the candidate solutions in following sections.

### 6.3.5 Estimation of various Miscellaneous Routes

The mainframe organization must estimate various miscellaneous routes which results in good load balancing and uphold connectivity even in case of different failure layouts. Estimating routes for Case-Debased and Case-Autonomous bifurcation is NP-hard, so use the group of routes estimated by ideal solution that optimizes for failure state separately. This ensures that the routes are adequately diverse to assure traffic delivery in all failure states, while also making efficient use of network resources. The ideal solution has an independent set of routes and bifurcation ratios in failure state $gs$. To circumvent having variables for exponentially many paths, the problem is defined in terms of amount of flow $TF_a^{gs}$ from request $q$ over edge ‘$a$’ for failure state ‘$gs$’. The ideal edge loads are obtained by solving the following equation:
The constraints of equation-3 are as follows: the first constraint represents total flow on edge ‘a’. The second constraint establishes flow maintenance. Third and Fourth constraints establish that the requests are met and last constraint assures flow non-negativity.

After getting the ideal flow on each edge for all the failure layouts, a classic dissipation algorithm is used. This algorithm will determine the similar routes $R_q$ and traffic $T_r^{gs}$ on each of them.

The algorithm is as follows:

1) Start with a set $R_q$ which is empty.

2) Add new exclusive routes to set $R_q$ executing following steps for each failure state ‘gs’ :-

2. A) define each edge ‘a’ with value $TF_{a,q}^{gs}$

2. B) Withdraw all edges that have ‘o’ value.

2. C) Obtain a route connecting $x^q$ and $y^q$.

Preferably select the shortest route between $x^q$ and $y^q$.

2. D) Add this route $r$ to the set $R_q$ and designate to it traffic $T_r^{gs}$ equal to the smallest value of the edges on route $r$.

3) Repeat the above steps until there are no edges left in graph.

Note that it can be shown by induction that this algorithm completely distributes the flow $TF_{a,q}^{gs}$ into routes. The dissipation yields at most $|E|$ routes for each network failure state $gs$ because the value of at least one edge becomes 0 whenever new route is discovered. Hence
the total size of set $R_q$ is at most $|E||GS|$. In practice, the above algorithm yields in smaller number of routes between each pair of edge routers.

6.4 Estimation of Traffic Bifurcation Range

Once the routes are determined, the network mainframe organization can now determine the route bifurcation range for each entering-exiting router pair. The optimization problem and resulting solution depend on whether the routers perform Case-Debased or Case-Autonomous bifurcation.

6.4.1 Case-Debased Bifurcation

In this type of bifurcation, each entering router $x^q$ has a set of bifurcation ranges for each tangible failure state $tg \in TG_q$. Since route bifurcation range depend on which routes in $R_q$ have failed, the entering router must save bifurcation range for min $(|GS|, 2^{|R_q|})$ layouts. Luckily, the number of route $|R_q|$ is typically small in practice. When network implement Case-Debased bifurcation, the mainframe organization’s task is to routes $R_q$ for each request and the traffic $T_{r}^{tg}$ on these routes in all tangible states $tg \in TG_q$ If the routes $R_q$ are known and stable, the problem can be represented as:

\[
\text{Min obj } \left( T_{F_a}^{gs} / M^{a1} \right) \\
\text{Sub to} \\
T_{F_a}^{gs} = \sum_{q \in RQ} \sum_{\text{routes} r} T_{r}^{tg} \quad \forall \: gs, a, tg = t_g(a, gs) \\
b^q = \sum_{\text{routes} r} T_{r}^{tg} \quad \forall \: q, tg = TG_q \\
0 \leq T_{r}^{tg}
\]

The constraints of equation 4 are as follows: the first constraint represents total flow on edge ‘a’. Second constraint ensures that request $q$ is fulfilled in all tangible failure states and last constraint assures non-negativity of traffic.
6.4.2 Case-Autonomous Bifurcation

In this type of bifurcation each entering router has a single anatomy entry which contains the bifurcation range that is used under any combination of route failures. Each router ‘r’ is related with a bifurcation fraction $\beta_r$. When one or more route fail, the entering router $x^q$ examine state $tg$ and uses equation 5:

$$\frac{\beta_r}{\sum_{c\in R_q} \beta_c}$$

as the bifurcation range for route r. If network implements Case-Autonomous bifurcation and the route $R_q$ are known and steady then problem can be represented as:

Min obj ($T_{s}^{gs1}/M^{a1}$)

Sub to

$$T_r^{tg} = b^q \frac{\beta_r}{\sum_{c\in R_q} \beta_c} \quad \forall \, q, \, tg = TG_q, \, r \in R_q$$

$T_{a}^{gs} = \sum_{q\in RQ} \sum_{r\in R_q} \sum_{s\in R} T_r^{tg} \quad \forall \, g, \, a, \, tg = t g_q(g s)$

$0 \leq T_r^{tg} \quad \forall \, q, \, tg = TG_q, \, r \in R_q$

The constraints of equation 6 are as follows: the first constraint assures that the traffic accredited to every accessible route ‘r’ is proportional to $\beta_r$, other three constraints are same as in equation 4. Unluckily, no recognized optimization technique permits to estimate an optimal solution efficiently, even when the routes $R_q$ are fixed. Hence heuristics are used to find both the candidate routes $R_q$ and bifurcation range $\beta_r$

To find the set of routes $R_q$ use the ideal routes retrieved by dissipating equation (4). To find the bifurcation range, let

$$\beta_r = \sum_{gs \in GS} \frac{W^{gs} T_r^{gs}}{\beta_r}$$

129
Where $T_r^{gs}$ is the traffic designated by ideal solution to route ‘r’ in network failure state ‘gs’. Since $\sum W^{gs} = 1$, the calculated range is the weighted average of the bifurcation range used by the ideal solution in equation (4).

6.5 Experimental Setup

To solve the equations 4 and 6 a simulator is developed that calls CPLEX [101] linear program solver in AMPL [98].

6.5.1 Empirical Plan

The simulation uses various kinds of topologies such as synthetic, Abilene [102] and Tier-1 IP backbone [100].

1. Synthetic topologies: It encompasses two-level hierarchical graphs, purely random graphs and Waxman graphs. These are generated using topology generation tool GT-ITM [27] and extended nam editor. The parameters used by these topologies are as follows: The hierarchical topology uses 50 nodes, 148 & 212 edges in HTa & HTb respectively with 2450 requests. Random topology uses 50 nodes, 228 & 245 edges in RTa & RTb respectively with 2450 requests. The waxman topology uses 50 nodes, 168 & 230 edges in WTa & WTb respectively with 2450 requests.

2. Abilene topology: The topology of the Abilene and a measured traffic matrix is used with edge magnitude of 10 Gps. Abilene topology use 11 nodes, 28 edges with 110 requests.

3. Tier-1 IP backbone: The city-level IP backbone of tier-1 ISP is used. Tier-1 topology uses 50 nodes, 180 edges and 625 requests.

Table 6.1 and 6.2 shows the datasets used in Abilene topology and tier-1 IP backbone. Format of the table can be explained as follows:

Each file contains $12 \times 24 \times 7 = 2016$ 5-min traffic matrices.

Unit: (100 bytes / 5 minutes) // 100 is the pkt sampling rate

Each line belongs to one TM; and each line contains $144 \times 5 = 720$ values organized as follows:

<realOD_1>\"
where simpleGravity means the independence model: \( p(s,d) = p(s)p(d) \), whereas generalGravity means the conditional independence model, which treats outbound traffic differently. simpleTomogravity and generalTomogravity are the tomogravity estimates with either simpleGravity or generalGravity as the prior.
<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.9570870e+06</td>
</tr>
<tr>
<td>7.8041337e+06</td>
</tr>
<tr>
<td>5.7008490e+06</td>
</tr>
<tr>
<td>4.6001000e+06</td>
</tr>
<tr>
<td>4.3011921e+06</td>
</tr>
<tr>
<td>7.4188240e+06</td>
</tr>
<tr>
<td>1.7065062e+06</td>
</tr>
<tr>
<td>2.5209508e+06</td>
</tr>
<tr>
<td>2.9506990e+06</td>
</tr>
<tr>
<td>3.4797676e+06</td>
</tr>
<tr>
<td>2.4407603e+07</td>
</tr>
<tr>
<td>3.7892983e+06</td>
</tr>
<tr>
<td>3.8256680e+06</td>
</tr>
<tr>
<td>7.8585035e+06</td>
</tr>
<tr>
<td>7.6934890e+06</td>
</tr>
<tr>
<td>1.6200030e+06</td>
</tr>
<tr>
<td>1.1945593e+07</td>
</tr>
<tr>
<td>9.4951129e+06</td>
</tr>
<tr>
<td>5.8337291e+06</td>
</tr>
<tr>
<td>6.3159457e+06</td>
</tr>
<tr>
<td>1.6004500e+05</td>
</tr>
<tr>
<td>6.7269923e+05</td>
</tr>
<tr>
<td>7.4761286e+05</td>
</tr>
<tr>
<td>4.5329386e+05</td>
</tr>
<tr>
<td>5.3978370e+05</td>
</tr>
<tr>
<td>7.339390e+06</td>
</tr>
<tr>
<td>7.9521740e+06</td>
</tr>
<tr>
<td>8.2362520e+06</td>
</tr>
<tr>
<td>4.8964027e+06</td>
</tr>
<tr>
<td>5.4405688e+06</td>
</tr>
</tbody>
</table>

**Table 6.1 Data set used in Abilene topology**

For details refer Appendix B.
Table 6.2 Data set used in tier-1 topology

For details refer Appendix B

The cluster of network failure GS for the synthetic topologies and Abilene comprise of single edge failures and no failure case. Two experiments with altering cluster of failures are executed on the tier-1 IP backbone:

a) First experiment uses single edge failures.
b) Second experiment uses cluster of failures which also contains common risk link cluster.

Along this a set of failures are also used. Value for common risk link cluster is procured from network operator’s database that contains 573 failures with the largest failure affecting 16 links simultaneously. The weight $W^{qs}$ in eq1 is set to 0.6 for no failure case and all other failure weights are equal and sum to 0.6.

All experiments are performed several times by increasing traffic requests constantly. In synthetic topology initial value for requests is the elementary value of requests which is then increased up to twice the initial values. In Abilene and tier-1 topology traffic is increased up to three times the initial value.

The $\phi$ [13] is a penalty function which helps in solving eq2 & eq3. It is defined as piecewise linear penalty function $\phi=0$ and its derivatives:

\[
\phi'(l) = \begin{cases} 
1 & \text{for } 0 \leq l < 0.33 \\
3 & \text{for } 0.333 \leq l < 0.667 \\
10 & \text{for } 0.667 \leq l < 0.9 \\
70 & \text{for } 0.9 \leq l < 1 \\
50 & \text{for } 1 \leq l < 1.1 \\
5000 & \text{for } 1.1 \leq l < \infty
\end{cases}
\]

The function can be viewed as modeling transmission delays caused by packet losses.

Simulation computes the objective value of the ideal solution, Case-Debased, Case-Autonomous and equal bifurcation. Equal bifurcation is modified Case-Autonomous bifurcation that bifurcate the traffic evenly on available routes. Simulations performed also compute the objective acquired by the shortest path routing of BGP.

6.5.2 Realizations with Stable Traffic

The main objective of traffic engineering is to abstain congestion and packet failure during the different failure scenarios. The objective as a function of the ascending request is shown in fig 6.1. The results which were obtained on hierarchal, Abilene and tier-1 topologies are representative; similar observations were made for all other topologies.
The graph in fig 6.1 shows the performance of Case-Debased bifurcation and ideal bifurcation. The difference between these two is negligible. Case-Autonomous bifurcation is disillusioned and it does not allow characteristic load balancing ratios for particular failures. Hence it has poor performance than ideal bifurcation. Equal bifurcation has worst performance.

Fig 6.1 The traffic engineering objective as a function of increasing traffic load in Hierarchical topology

Fig 6.2 The traffic engineering objective as a function of an increasing traffic load in Abilene topology
From fig 6.1, 6.2, 6.3(a) and 6.3(b) it can be observed that OSPF achieves a considerably worse performance than Case-Autonomous and Case-Debased bifurcation as the load increases. But it should be noted that in OSPF, each router is bounded to transmit all of its traffic on a single route with the smallest weight or bifurcate the traffic evenly if many
smallest-weight routes are available. This method does not permit the same flexibility in selecting routes and bifurcation ratios as permitted by proposed solution. Hence OSPF should not be predicted to achieve the same performance even for optimal choice of OSPF link weights [12].

Solutions with lesser number of routes are favoured because they lessen the number of channels that have to be managed and also reduce the size of router anatomy. But there should be enough routes available which can evade failures and lessen congestion. Notice that number of routes used by proposed algorithm is small. Fig 6.4(a) shows the number of routes used by each request.

Fig 6.4 (a) shows that number of routes increases as topologies become larger and discrete. Hierarchical topology uses 8 or fewer routes for 93% of requests. Fig 6.4 (b) shows that tier-1 topology uses less than 11 routes for all requests. It also shows that as the network flow is increased, there is mild increase in number of routes. This increase is caused by moving some traffic to longer routes as shorter routes become congested.

![Fig 6.4 (a) number of paths used by various topologies](image-url)
Fig 6.4 (b) Number of paths in tier-1 topology with cluster risk links

Fig 6.5 Size of compressed routing table
A realistic solution utilizes less MPLS [1] labels in order to lessen the size of routing tables in routers. Experimental results show that when MPLS tunnels are used in tier-1 topology, a few thousand tunnels can pass through single router. But the elementary routing table compression techniques allow reducing size of routing table to few hundred entries in each router. Such compression is substantial as it reduces the memory requirements imposed on elementary routers and it also improves the route lookup time.

Routing can be compressed by utilizing same MPLS labels for routes with common path to destination. Particularly, if two routes to destination \( z \) traverse through router \( r \) and these routes use the same path between router \( r \) and destination \( z \), then same outbound label should be used in routing table of router \( r \). Fig 6.5 depicts routing table sizes as a function of network load. The arc on the top shows the size of largest routing table and arc on the bottom shows the average routing table size among all the backbone routers.

### 6.5.3 Realizations with Progressive Traffic

As the traffic matrix changes it becomes inconvenient to solve equation 4 and equation 6 because due to change in traffic matrix it is needed to alter to router anatomy with new updated routes and new bifurcation ranges. The solution to this problem is use a single router anatomy which is vigorous to periodic changes of the requests. For this, netflow traffic traces is collected for a single day and is processed using NFSen [64] tool. Fig 6.7 shows the collective traffic volume along with traffic between three entering-exiting router pairs. From graph it is clearly visible that at 9 am traffic volume reduces to its lowest and it reaches to its maximum at midnight and 8pm. To estimate single set of routes that ensures failure flexibility and load balancing for each of the 24 traffic matrices the following method is used:

a) Wrap 24 traffic matrices instead of using their average i.e. let \( TM_{ij} = \max_k TM_{ij}^k \).

b) Now calculate router anatomy robust to traffic changes.

c) Testing is performed on solution by changing traffic request.
Fig 6.6 Combined traffic volume in the tier-1 network.

Fig 6.7 The traffic engineering objective in tier-1 topology
Fig 6.8 shows the objective value of Case-Debased bifurcation and Case-Autonomous bifurcation. This shows that Case-Debased bifurcation achieves same result as the ideal bifurcation and is robust to periodic traffic changes. Case-Autonomous also achieves same result as ideal bifurcation but it deviates when traffic increases during peak hours.

6.6 Implementation Layouts

Despite the proposed framework allows the use of new simpler routes but this framework can be implemented with current protocols and equipments. An ISP can implement proposed framework using MPLS [26]. Data centers can utilize same solution or they can have assistance from current Ethernet switches and transfer some functionality in to end-host machines.

6.6.1 Utilization of MPLS in ISP backbone

- Establishing MPLS routes with RSVP (resource reservation protocol): MPLS is specifically appropriate because entering routers enclose packets with labels and move them over pre-estimated label switched paths (LSPs). This allows supple routing when various LSPs are established between entering-exit router pair. The proposed framework can be viewed as a specific application of MPLS, where mainframe organization calculates the LSPs and notify the entering routers to establish paths using RSVP and attenuate any progressive recalculation of alternate route when primary route fails.

- Bifurcation based on hash: multipath forwarding is held up by commercial routers of both major vendors. The routers can be arranged to hash packets depend on port and address information in the headers into various groups and forward each group on a different path [25, 49]. This supports route bifurcation with nearly fine granularity,
while avoiding out-of-order packet delivery by assuring that packets associated to same TCP or UDP flow moves through same path.

- Failure tracking using BFD: fast failure tracking can be done using bidirectional forwarding detection (BFD). A BFD session can supervise each route between two routers, by piggybacking on the current traffic flow [19].

- Failure restoration: the entering router adjusts to path failures by bifurcation of traffic over the remaining paths. In Case-Autonomous bifurcation, the entering router has a single set of traffic-bifurcation weights and automatically renormalizes to direct traffic over working paths. Case-Debased bifurcation needs alteration to router software to move alternate traffic-bifurcation weights in data-plane, no hardware alteration is needed.

- Estimating traffic requests: MPLS has SNMP (simple network management protocol) counters (known as management information base (MIB)) that estimate the total traffic moving through each label switched path. The management system can poll these counters to estimate the traffic requests otherwise NetFlow or tomography can also be used.

### 6.6.2 Utilization of hosts and switches by data centres

While data centre can easily utilize the same MPLS-based solution but there is another solution also available which is based on the use of end-host and cheaper switches.

- End-host assistance for supervising and traffic bifurcation: The server machines in data centres can achieve many path-level operations. For example in VL2 [4] and SPAIN [43] framework, the end-host can enclose the packets to move them towards a particular path. This allows fine-grain traffic bifurcation. In addition, the end host can carry out path level investigation in data plane, by piggybacking on current traffic flow and send extra active probes when required. Upon tracing path failures, the end-host can alter to new path bifurcation percentages which are based on the pre-estimated anatomy installed by controller. The end-host can also estimate the traffic requests by keeping counts of traffic directed to each exit switch.

- Switches: The leftover functions can be performed by the underlying switches. For example: the mainframe organization can set up multiple paths by combining these paths into set of trees, where each tree correlates to different VLAN [43]. If switches have assistance of emerging OpenFlow [63, 65] standard, then mainframe organization can install a forwarding table rule for each hop in each path, where rule
matches on the VLAN tag and forwards the packet to appropriate output port. Since OpenFlow switches maintain traffic counters, mainframe organization can measure traffic requests.